

Sketching designs using the Five Design-Sheet methodology

Jonathan C. Roberts, *Member, IEEE*, Chris Headleand *Member, IEEE*, and Panagiotis D. Ritsos, *Member, IEEE*

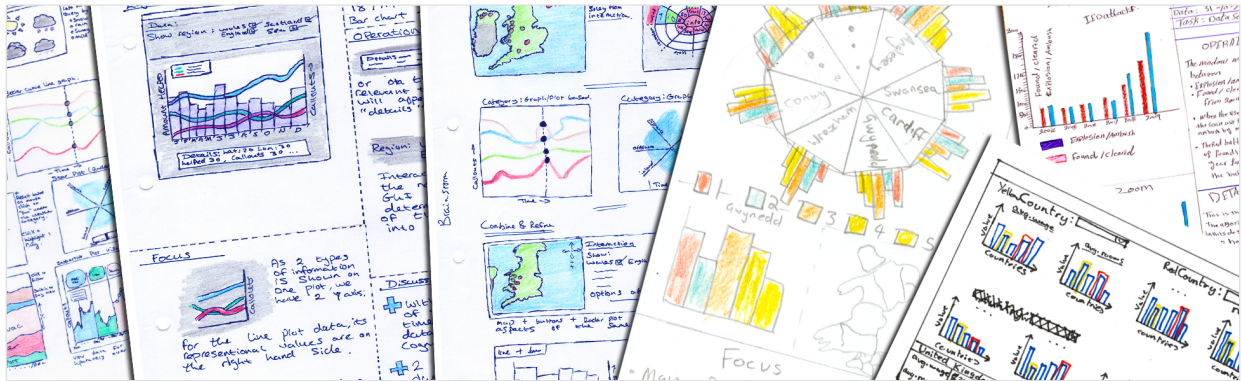


Fig. 1: The FdS consists of five sheets, each sheet with five parts. It is designed to help users consider alternative visualization designs; plan designs and create a realization sheet that can then be coded. Pictures of students' FdS sheets are shown.

Abstract— Sketching designs has been shown to be a useful way of planning and considering alternative solutions. The use of lo-fidelity prototyping, especially paper-based sketching, can save time, money and converge to better solutions more quickly. However, this design process is often viewed to be too informal. Consequently users do not know how to manage their thoughts and ideas (to first think divergently, to then finally converge on a suitable solution). We present the Five Design Sheet (FdS) methodology. The methodology enables users to create information visualization interfaces through lo-fidelity methods. Users sketch and plan their ideas, helping them express different possibilities, think through these ideas to consider their potential effectiveness as solutions to the task (sheet 1); they create three principle designs (sheets 2,3 and 4); before converging on a final realization design that can then be implemented (sheet 5). In this article, we present (i) a review of the use of sketching as a planning method for visualization and the benefits of sketching, (ii) a detailed description of the Five Design Sheet (FdS) methodology, and (iii) an evaluation of the FdS using the System Usability Scale, along with a case-study of its use in industry and experience of its use in teaching.

Index Terms—Lo-fidelity prototyping, User-centred design, Sketching for visualization, Ideation

1 INTRODUCTION

One of the challenges for developers wishing to create a new information visualization tool, is deciding how they want to visualize the data. Coming up with novel visualization forms is difficult; the visualization designers need to think divergently and consider different possibilities. It is a process that may not come naturally. Conceiving a variety of novel visualization concepts is an exercise in divergent thinking, a thought process that is used to generate a variety of creative ideas.

Good ideas can be conceived through generating and evaluating many possible design solutions. Linus Pauling (the Nobel prize winning chemist) said “*The way to get good ideas is to get lots of ideas, and throw the bad ones away*” [12]. Creating these different ideas means that the developer needs a quick way to capture individual concepts, in order to facilitate the selection and improvement of different design schema. Specifically, this approach has two main challenges: (i) how to conceive and stimulate the production of novel ideas, and

(ii) how to manage the ideation process, converging to a single realizable design.

Sketching and lo-fidelity prototyping has been shown to enable individuals to record such ideas quickly [7, 11, 13]. Consequently, sketching can help organize thoughts, and make explicit and externalize the possible ideas in their mind. The sketches produced in this planning would then be turned into the interface, or the underlying algorithm. Sketching interfaces has long been used in software engineering. For instance, wireframe designs have become an increasingly popular tool, in the recent years, for the production of web interfaces. However, while such techniques are good to illustrate individual interfaces, they do not help manage their creation process, nor do capture the underpinning dependency on data when used for information visualization applications. For that, what is required is a strategy that is appropriate for managing the sketching of designs suitable for information visualization and facilitating the decision making process.

We present the Five Design Sheet methodology (FdS): a complete solution that fosters divergent thinking, is based on sketching, and enables users to contemplate on the goal of the tool to be designed, its operations and interaction. The FdS has been devised with developers, data scientists and visualization educators in mind (referred-to collectively as ‘users’). The methodology consists of five sheets, each of which has five parts. It provides a formal structure, is focused on creating an interactive interface and encourages the user to think creatively.

The Five Design Sheet methodology has been used by students, designers, and companies for over five years. We first publicly outlined the FdS in 2011 at the Eurographics education stream [36] based on initial experiences with the method. This article substantially expands that conference paper. We provide more details of the methodology,

- Jonathan C. Roberts is with the School of Computer Science, Bangor University. E-mail: j.c.roberts@bangor.ac.uk
- Chris Headleand is with the School of Computer Science, Bangor University. E-mail: c.headleand@bangor.ac.uk
- Panagiotis D. Ritsos is with the Department of Computer Science, University of Chester. E-mail: p.ritsos@chester.ac.uk
- Resources are stored at <http://fds.design>

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication xx Aug. 2015; date of current version 25 Oct. 2015.

For information on obtaining reprints of this article, please send e-mail to: tvccg@computer.org.

a deeper reflection on sketching, and our experiences of its use and application over the past five years. In fact, within Bangor University alone it has been used by students studying for their individual third year (undergraduate) computer science project; on an Information Visualization module; in MSc projects and research by doctoral candidates. It has been used at three Visual Analytic Summer Schools (2010, 12 and 2013), and by a number of companies to consult with clients and develop new interactive solutions.

The goal of this article is to describe the FdS model in detail and provide a reflection on its use over a five year period. This case study enables us to present how effective the FdS has been in several contexts (as a teaching aid, in a commercial setting and for workshops). We feel that this paper is relevant and interesting to a wide range of readers. From learners wishing to plan to create a new visualization tool; teachers who wish to utilize and teach a design model to their students; developers who wish to create new designs and share them with clients or use them at design meetings; companies who wish to design and create interactive applications. Our contributions are threefold:

1. Thorough review of related work: visualization design and sketching as a planning method for visualization (Sec. 2).
2. Detailed description of the Five Design Sheet (FdS) methodology for visual design (Sec. 3), including an appraisal of different facets of sketching to facilitate the ideation process.
3. Evaluation of the FdS from three complimentary perspectives
 - (a) Evaluation of the FdS using the System Usability Scale [5] (SUS) (Sec. 4.1);
 - (b) Case study of FdS' use in industry; we explain the designs and how it saved time and money (Sec. 4.2);
 - (c) Presentation of the use of FdS in teaching – we present results of a case study, that describes students' experience using FdS for a postgraduate information visualization module over a five year period (Sec. 4.3);

2 RELATED WORK

The FdS is a methodology that encourages creativity and structures the ideation process. It guides users, to create different designs, through sketching on pieces of paper. The goal of this process is that the user creates a new visualization design. It provides a useful *interaction design* methodology [39] because it involves clients at each stage of the development to create useful designs. Consequently, we divide the related work into two parts: first, design methodologies for visualization, and second sketching as a planning tool for visualization design.

2.1 Visualization design

Most design models follow a *design study* approach [41], where researchers along with domain experts tackle a real-world challenge, prototype a solution and then reflect on their work. However, how do these ideas get created in the first place? How does a user run through a set of initial possible alternatives? It is within this early 'idea space' and conceptualization stage that the FdS works well.

Munzner [30] presents a nested model for visualization design. Her model consists of four tasks: domain and the characterization of the problem, followed by operation abstraction design, interaction design and finally algorithm design. Our FdS methodology focuses on the first three tasks in Munzner's model. Users categorize the problem in sheet 1, whereas the remaining four sheets enable users to think through the interface, operations and algorithms, in separate subsections of each sheet. Munzner argues that her model can be applied in both problem-driven and technique-driven research and is complementary to recommendations from other researchers, (e.g., [50]). Likewise the FdS can be applied to problem and technique-driven research strategies. McKenna et al. [29] propose the Design Activity Framework based on Munzner's nested model, where users Understand, Ideate, Make and Deploy (UIMD). The model is used as part of a wider project plan with several deadlines throughout the whole project.

Roam [34] presents his SQUID method as a way to encourage users to think about the problem in different ways; ie., simple vs. elaborate;

qualitative vs. quantitative, vision and execution, individual and comparison, difference and 'as is'. In particular, he presents a visual language, in which if we want to express 'who' or 'what' we draw a portrait, whereas to visualize quantity we should draw a chart. This idea can be used to plan visualization design. However, while this method is good to articulate ideas, it does not focus on interface design.

There are other models that help users think about the visualization mapping, but do not help users develop creative interfaces. For instance, Bertin invites us to consider the components of the system, where users should ascertain the data components and then work out the components of the retinal system, and to map one to the other [4]; Spence [44] identifies selection, encoding and presentation as sub-steps in the raw data transformation; while Ware's model [53] includes data collection and storage, preprocessing of the data, display algorithms for presenting the information and, finally, the human perceptual and cognitive system. There are other visualization models that help users conceptualize the whole process. For instance, dataflow [49], data state [9], p-set for exploration [24], lattice [22], sensemaking [31] and data frame [27] all help the developer divide the program into separate components (or objects). While *visualization patterns* specifically help users develop code for visualization through a pattern-based approach (see [20] and [45]), higher level models also can help the user frame the idea space (e.g., [8, 32]). However these models do not help users to think through ideas in a initial concept stage. We need to look to other design fields for inspiration for models and guidance.

Participatory designing is a method often used in product design, where groups of users come together to ideate and outline solutions. While it is unclear whether working together, in this way, does actually create new ideas [54] – with people being cautious of engaging, embarrassed, or worried their ideas would be copied, or bullied by a strong person – certainly working with clients over a problem does help [26].

Linus Pauling's advice is to consider lots of different ideas and sift through them for their appropriateness [12]. This twofold strategy can be expressed as applying *divergent* and *convergent* thinking. Users think of many solutions. All ideas are feasible, it does not matter if they are workable or impossible, what matters is that the user creates lots of them. Then at a later stage the user starts to reduce down these possible solutions, and critically analyze them to decide whether they are in fact possible and doable. Divergent thinking is about broadening the possibilities; while convergent thinking is to focus on specific solutions with specific answers.

The Design Council's *double diamond* [10] (created after analyzing the design process of 11 companies) places this strategy in a broader context, as divergence and convergences can occur at different parts of the design process. Users Discover, Define, Develop and Deliver the product. Importantly the double-diamond model adds an agile approach to the *develop* and *deliver* stages, where prototypes are built and tested for their effectiveness before being developed further. This is important for our FdS strategy, because we also encourage users to create designs and further refine their ideas through iteration. Such agile iteration, and refinement of ideas, is indeed a recurring theme of many of these design papers. For instance, Shneiderman [43] writing on his Genex framework (generator of excellence) – that considers four core areas *Collect, Relate, Create and Donate* – "*These four phases are not a linear path ... creative work may require returning to earlier phases and much iteration*". Shneiderman's Genex is similar to the model of Sanders and Stappers [40] (SS). Users collect data, synthesize ideas, explore different possible solutions then prototype the answers. Our FdS methodology fits with Double Diamond, Genex and SS (see Fig. 2).

In fact, we believe that this refinement process places the user on a journey; they think up new ideas, develop improvements and revise their assumptions before a final design is reached. Through this 'idea exploration' the user hopefully converges onto a solution that fits the specification and requirements. The developer should be encouraged to consider novel and unusual solutions; perhaps to push the boundary of all possible ideas, but also to help ensure a 'best-fit' solution.

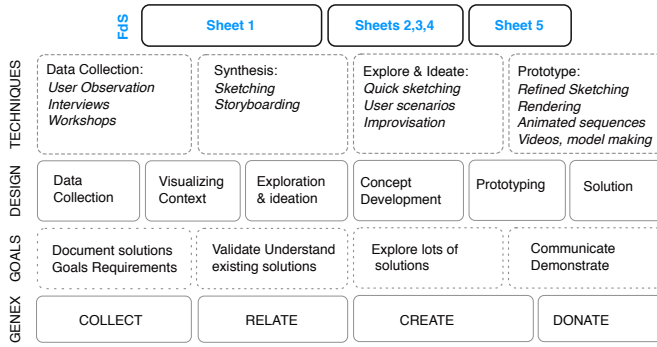


Fig. 2: Schematic that shows where the FdS design fits in with the Genex model of Shneiderman [43] (collect, relate, donate and create) and the design process of Sanders and Stappers' [40].

These ideas would certainly adapt and be improved at later stages of the design process. Nonetheless, the goal of the 'early' process is exploratory. In fact, for the FdS we are not concerned with data collection, but users do need to think about the data, and to consider, synthesize and consolidate ideas in sheet 1. Users need to think over the data and to consider the different parts of the data at this stage. They explore different possible solutions (sheets 2,3 and 4) and finally plan a prototype.

2.2 Sketching as a planning method for visualization

Many creative industries use sketching as a way to investigate, explore and plan different possible solutions. E.g., product, fashion and graphic designers, architects and film-makers all sketch many different possible solutions. Heller and Landers provide insights into about fifty designers' sketching practices based on excerpts of their sketchbooks [21]. The use of lo-fidelity sketching frees the user from worrying about technical limitations or assumptions and encourages them to explore different solutions. In fact architectural design was one of the main inspirations for our work. Tovey writes "[designers use sketches to] generate concepts, to externalize and visualize problems, to facilitate problem solving and creative effort, revising and refining ideas" [46]. In visualization, this has been less formally used. Users often sketch and plan, but usually don't follow a method, rather they do it in an ad hoc way. Recent work by Keefe [26] and Jackson et al. [23] demonstrate the power of sketching; they explore one designer generating several solutions, and make comparison to other lo-fidelity prototyping methods. Sketching is also used by Walny et al. [52], where users directly sketch the data.

Another inspirational idea from architecture design was the idea of the *parti pris* [17] (the big idea). The word comes from the French *prendre parti*, a bias or a mind-made-up. In architectural-criticism the *parti* is an assumption that informs the design; it is therefore the central, most overarching concept that the design is portraying. In other words, it is pivotal to making the design work. Let's consider the example of a parallel coordinate plot. In this case the *parti* is the fact that axes are parallel and the data is plotted as polylines across the axis. Each sheet of the FdS (apart from the first) have a focus/parti segment.

Rettig [33] writes, "Lo-fi prototyping works because it effectively educates developers to have a concern for usability and formative evaluation, and because it maximizes the number of times you get to refine your design before you must commit to code". He encourages iteration and refinement at the prototype stage, saying that quality of the final product comes through iterative refinement: "get the big things right during lo-fi, and the little things will follow in future iterations". Rettig gave users a pragmatic set of instructions for programmers to develop lo-fi prototypes on paper: assemble a kit (pens, paper, ruler, scissors, etc.), set a deadline, draw models not illustrations. He also suggests that one sheet of paper should be used per interface. Then these paper interfaces could be tested with users. So, prepare the test,

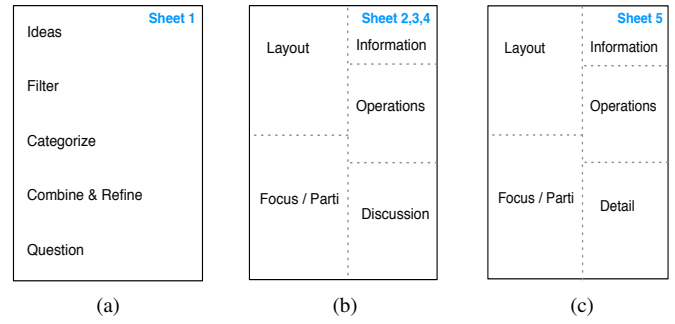


Fig. 3: The FdS sheets. (a) Sheet 1: Generate Ideas, filter, categorize, combine & refine then question. (b) Sheets 2,3,4 with the five sections in the 2-row 3-row format; (c) Sheet 5, the realization sheet where *Detail* is included instead of Discussion.

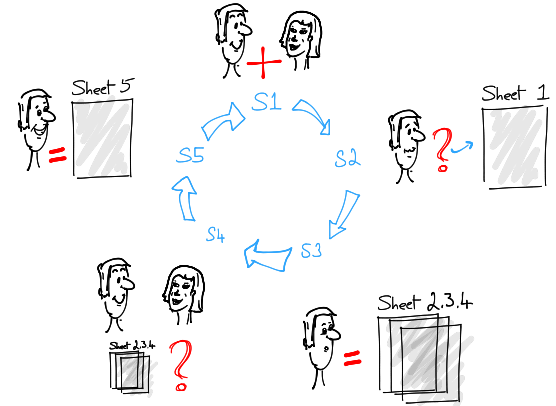


Fig. 4: Five stages to the FdS: (S1) meet with client and consider task; or contemplate task on own. (S2) Ideate and sketch small ideas. (S3) Sketch and plan three alternative designs. (S4) Consider solutions with client; or deliberate on own. (S5) Generate realization sheet, and implement prototype. Discuss with client and re-iterate if necessary.

select users, prepare test scenarios, practice these scenarios, and allocate roles (greeter, facilitator, computer, observers).

Our focus on sketching fits well with other work in the visualization domain. For instance, Craft and Cairns [11] and Curtis and Vertelney [13] encourage storyboarding and sketching prototypes for rapid visualization interface development. Roam [34] presents a series of visual sketching methods as a way to solve problems in business and help developers crystallize ideas. Buxton et al. [7] encourage sketching for interface design.

Through sketching the design is recorded, and tells the story of the fluid, ephemeral evolution of the idea [3]. Users often sketch multiple designs on the same sheet of paper [18]. Even when the designer uses a computer to create different 3D models, they often render the output in a sketchy appearance. Similarly prototype visualization tools can be rendered in a sketchy appearance (e.g., [28, 55]) while sketching can also be an input device [42].

3 THE FdS METHODOLOGY

The FdS is a five-stage methodology (Fig. 4) comprising of five sheets (Fig. 3), each sheet containing five parts. Explicitly, the first sheet is the brainstorm (ideas) sheet (Fig. 3a); three design sheets (Fig. 3b) and a realization sheet (Fig. 3c). The latter four sheets are similar in construction. The methodology is summarized as follows:

1. **Five stages.** The whole process consists of five stages, (see Fig. 4). (1) the user considers the task (the user meets the client). (2) The user thinks divergently and considers many alternative ideas.

- (3) The user creates three principle designs. (4) The designs are considered, reflected upon, or discussed with the client. (5) A final design (the realization sheet) is created.
- Five sheets.** Sheet 1 allows users to explore ideas and to sketch many different concepts; sheet 2, 3 and 4 are the main design sheets; with sheet 5 being the realization sheet (see Fig. 3 and Sections 3.4 and 3.5).
 - Five parts to brainstorm.** On sheet 1, there are five sections to help divergent thought: Ideas, Filter, Categorize, Combine & Refine, and Question (detailed in Section 3.3).
 - Five parts to each design sheet (2,3,4) and the realization sheet 5.** Each principle design sheet contains five sections in a 2 row by 3 row grid.

The intended users of the FdS are designers, creating plans for someone else to implement, and developers who are planning to code a visualization tool. Either of them can be individuals, or working in industry for a company, who wish to visualize a dataset. Alternatively they can be learners, and instructors who are helping students to learn how to consider alternative design solutions and plan to code.

An example of the five FdS sheets is shown in Fig. 6. This work was produced for an assessment on the ICP4142 Information visualization module, for the MSc Computer Science program. The student chose data that recorded performance indicators of universities, and contained data on widening participation and disability support at different institutions. The student chose the dataset and followed the FdS methodology, and then developed a prototype tool in Java using the Processing.org library. The figure demonstrates how the student explored different potential visual depictions of the data, before finalizing a chosen design.

3.1 Facets of lo-fidelity sketching for Visualization Design

There are many reasons for using lo-fi sketching. Not only does it help users make explicit their thoughts, but it also helps them to share their thoughts with others and collaboratively work on a design. We consider sketching from five points of view: who the sketcher is, the physicality of the medium, the appearance of the creation, its purpose and implied dynamic nature (see Fig. 5).

The sketcher. The person sketching would normally be the developer, i.e., the person who will create the visualization tool. They may work alone and use the FdS to plan their own thoughts, or more usually create and share their ideas with a client, or at least discuss the ideas with another person. The act of discussing and explaining the sketched ideas helps the user to confirm and refine the concepts. Certainly not everyone likes to use pens and pencils, and some users say that they cannot draw. However these sketches are quick renderings of ideas, and are not intended to be final products. Therefore users can simply use geometric shapes, lines, rectangles and circles. Users should not spend undue time on the sketches; they are meant for a purpose and not works-of-art.

The sketching medium we are advocating is physical. Pens or pencils are natural interfaces (because we have learned to use to them through childhood). The resources are cheap, repeatable and widely available. They are at the right level to explore designs early and decide the ideas are good or bad [7]. Other physical medium, such as whiteboards and dry-wipe pens, may be suitable for some purposes, especially if groups of individuals are working together. It is possible to use wire-frame generation software, or drawing applications to create these designs, however, from our experience, these tools can restrict rather than engender the creative process. Users sometimes spend more time changing parameters and choosing operations on the application rather than being creative.

The appearance of the sketch gives the observer (user or client) a clear understanding that it is unfinished, and that it drafts some concept. This is beneficial in visualization design, because it evokes and enables discussion, without potential embarrassment

Sketching designs with the FdS

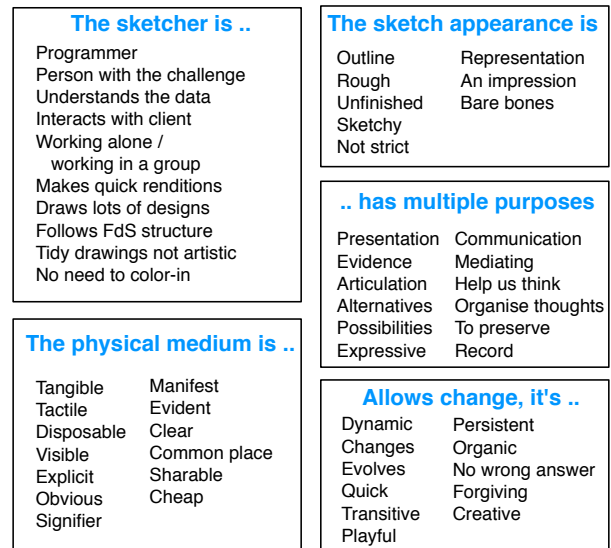


Fig. 5: Sketching is useful because it provides a way to explore several different ideas; especially articulate ideas that have not been fully formed in the user's mind. By making these thoughts explicit the user often firms up their ideas. Benefits come from its *physical medium*; unfinished and draft *appearance*; transitive and quick to create *nature*; and that there are different *purposes* to the sketch copy.

that the developer has wasted his/her time on implementing possible bad solutions. Even if the author does not believe that they can sketch, they are forgiving as the user can merely try another one, throw it away, or use simple shapes to express the core of their ideas.

The purpose. Sketches not only help users express their ideas in a way that can be exchanged with other people, but they also help users consolidate ideas. The sketching process helps users to make explicit their tacit and internal thoughts. The physical nature externalizes inner thoughts. The many pieces of paper created provide a record of the thought process (especially useful for teaching purposes, because then the teacher can grade the process as well as the final picture). They are physical persistent forms of communication; old versions can be kept as a record of how the current set of thoughts have evolved.

The nature of these sketches is dynamic: they are adapted and changed by authors as they develop new ideas or change their thoughts about previous design ideas. They are organic and develop as ideas evolve. They also enable users to be playful and to try-out different scenarios. These thoughts can even be unworkable, unfeasible or plainly wrong, but by expressing these ideas in sketch form the user is free to edit and change them. They can evolve and change as thoughts develop.

3.2 Preparation for the FdS

For the FdS we are not concerned with the collection of data, however users do need to think about the data and to consider the overall goals and motivation of the visualization tool. As preparation for using the FdS there are four aspects that need to be considered as a preamble:

- Data:** The FdS assumes the existence of data. If there is no data or the data is unavailable (e.g., for privacy reasons) then it is impossible to continue.
- Task:** The user needs to understand the task. Whether working with the client or working alone, the user needs to understand the reason for looking at the data, and the goal or outcome that they are trying to ascertain. Different tasks may imply different tools.
- Components:** The user needs to understand the components of the data. We encourage users to consider five aspects of the data:

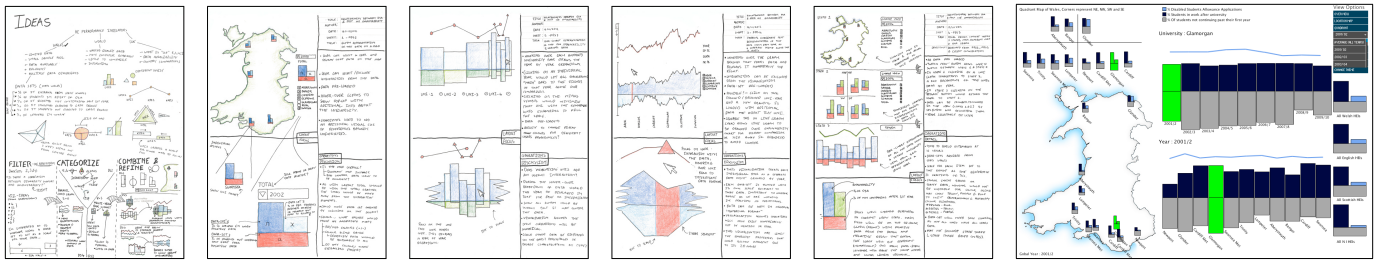


Fig. 6: An example of the FdS are shown on the left, with a picture of the final prototype on the right. Created for the Information Visualization module as part of the MSc course. The student chose to investigate data regarding University access for disabled students.

- Variables.** List the data variables. What are the parameters? Give them names. What are dependent and independent variables?
 - Types** of data. What data-types are in the data? How are they stored? What is the access to the data (API, JSON file etc.)? What data-structure holds the data (e.g., is it hierarchical)?
 - Categories.** Are the variables categorical (nominal or ordinal or ranks), are they quantitative (discrete or continuous)?
 - Temporal.** Is the data streaming data? How was it stored (all at one time or over several years)?
 - Range & distribution.** What is the distribution of the data? Few values, small size, evenly spread, sparse or dense?
4. **Resources.** Finally the user needs to gather resources together to create the FdS, e.g., colored pens or pencils, ruler and paper.

3.3 FdS Sheet 1: Ideation

Ideation is the process of creating new ideas. But where do ideas come from? How do users actually think up ten, twenty or a hundred ideas? How do new concepts get ‘born’? The five parts of sheet 1 leads the user to think divergently, to first generate *ideas*, then *filter* and *categorize* them, followed by *combining* and *refining* them, before finally *questioning* their suitability to the task, see Fig. 3

Ideas emerge by thinking, talking with other people, reading relevant literature, gaining inspiration from other domains, resting and not rushing, reflecting and collaborating. This theory is summarized in the following literature: **Relax:** Good ideas come through long, slow and careful thought, says Johnson [25] (he talks about a ‘slow hunch’). **Re-work:** Webb [57] says we should gather-material, think, relax and re-work. This reflection stage is extremely important in many domains (we often encourage our students to write their work and then leave it overnight before reading it again). **Provoke:** e.g., ask difficult questions, think of impossible solutions. **Persist:** Shneiderman [43] agrees and says it’s “1% inspiration and 99% perspiration”. **Iterate & Refine:** evaluate, revisit assumptions and re-design [10]. **Different personalities:** De Bono encourages us to take on different personalities in his six thinking hats [15]. **Collaborate:** Work with different people, with different skills and knowledge [48]. **Dissimilar ideas:** Glue dissimilar ideas together. E.g., Johnson [25] suggests finding dissimilar ideas and joining them together and through this joining up of different thoughts new ideas can be formed. de Bono’s ‘green-hat’ suggests using random pages of a book to inspire [14]. **Transference:** Look to other subjects for inspiration, e.g., biomimicry [38]. **Research:** Discover every idea and every solution so far. Look at other ideas and learn from others’ work [43]. **Metaphors:** Generate abstractions and use analogies. Metaphors help users to instantly understand the corresponding idea [38]. Ziemkiewicz and Kosara [58] suggest that metaphors work both ways: they both inspire and are needed for interpretation. **Make mistakes:** Good ideas can come from serendipity. Either try to fix these mistakes or use the result to your favor. There are well written examples where scientists make errors or have accidents and it is these that are actually good ideas: e.g., sticky-note glue or the discovery of penicillin [54]. **Reverse/Invert:** Reverse, flip or invert an

idea to generate others.

As the user goes through this exercise they need to think-through different possibilities, but also keep their mind on the task. Usually the user does each action in turn, and decides whether they have completed it to a satisfactory level before moving onto the next task. But the very nature of considering the next task may put them back to an earlier stage. E.g., by categorizing the ideas the user may realize that there is a missing category and thus move back to drawing more ideas. That themselves can be categorized. Indeed the point of each stage is not only divergent and convergent thinking, but also to provoke new ideas and increase the potential set of ideas.

- Ideate.** Users need to sketch as many ideas as possible. These are ‘mini-ideas’. They are lots of little drawings. While they could be full solutions, they are more likely to be ‘insufficiently thought out’ or ‘half-baked’ ideas, short concepts, or even wacky concepts. At the start, the point is to articulate different potential ideas. The thoughts should be driven by the task and the user should have ‘half an eye’ on the goal (to develop a tool that will visualize data), and should hold off criticizing the ideas because this will be done later. When a substantial¹ amount of ideas have been made the user moves on to *filter*.
- Filter.** Users should start to remove any duplicated ideas, or concepts that are irrelevant or absolutely impossible. Users can use these negative thoughts positively, such that if they are impossible then they should consider how they could they be fixed. In practice the sketches are being annotated rather than fully deleted (a single line crossed through a design will suffice). When considering these aspects, users need to think what is suitable to the task, and how new ideas can be generated from these deletions or duplications.
- Categorize.** Users need to consider what is similar and what is different. Annotation can be used to group similar ideas together. Categories change and develop, therefore users should not worry about the fine details of categorization: it is merely a tool to facilitate the exploration of ideas. While this operation converges (reduces) the designs, users should consider ‘what is missing’. E.g., is there another category of designs that should be present? What is this category? Is it relevant?
- Combine & refine.** Users need to organize the mini-ideas into bigger solutions. For instance, look to develop Multiple Coordinate Views [35]. Think what visualizations complement each other. E.g., spatial (map) with temporal (timeline), or overview with detail. Indeed the Space-time cube is an example of combining ideas [1]. Refine the ideas, making sure they use suitable colors [19] or other best practice [53]. Draw new combinations, look to refining or changing any designs; start to consider which three ideas will be planned in more detail on Sheets 2,3,4. Draw or annotate circles around three possible choices.
- Question.** Users should reflect on what has been created. Do solutions meet the task? Are they effective designs? Do they mis-

¹ It is difficult to quantify the number of mini-ideas required; but the more ideas that are sketched, the easier the next stages of filter, categorize etc. will be. Three is too few; 10 may be ok; 20 would be better.

lead or lie [47]. What are their advantages and disadvantages? Are there three or more designs? Users must consider which design is best and which three would be suitable to expand. Ideally, three totally different designs are required that represent alternative possible solutions, i.e., their *parti* sketch is dissimilar.

3.4 FdS Sheets 2,3,4: Alternative designs

The three design sheets enable the user to record detailed alternative concepts. The use of three sheets is for guidance. Too many designs can be confusing to a client and the user would need to spend more time expanding on each idea. Too few designs means that the possibility of alternative designs will not be explored. When the three designs are used in design concept mediation, the client is able to see all the main alternatives. Also, we encourage that completely different designs should be sketched on sheets 2,3 and 4. It may be that the user only considers two designs to be appropriate, but it would be suitable to add a third design to enable a full discussion with the client. The client also may see benefit and value to that idea, and a new concept may be born from it. In fact, selecting components from three designs leads to many possible combinations and a greater exploration of the design space. Any amendments and alterations, or fine detail can be added on the fifth sheet.

When crafting the three designs the developer should consider the appropriateness of the idea; how the solution matches the task; what the user interface would look like; and what algorithms could make this happen. The user should be less worried about implementation costs, or feasibility of the design, as these should be included in the *discussion* part of the sheet.

While a user can complete the five parts in any order, we advise them to be completed 1 through 5 (as listed below). Each of the panels relate to the other; in fact, we encourage users to label the elements in the Layout, and refer to these labels in the other panels. This makes most sense with Layout and Operations, because the location of the interface components is shown in Layout, explained in Operations, and expanded in Focus/Parti panel. We show the layout of sheets 2,3,4 in Fig. 3b and detail the parts below:

1. **Information.** Add appropriate meta-information, including title, author, date, task and sheet number.
2. **Layout.** This is the overview, a sketch the final interface, including buttons, GUI elements, visualizations, menus etc. It should look like a sketch of a screen-shot of the final application.
3. **Operations.** These are Action – Result pairs (\Rightarrow). What is required is a description of the interface-component and what happens when that component is activated. Multiple results could occur from that initial component activation, and different types of interaction can be described (from buttons, sliders to lasso selection and gestures). E.g., on release of a button click \Rightarrow the data is loaded, and is automatically visualized. Label these action/results such that they relate to Layout.
4. **Focus/Parti.** The Parti is the central idea, the core concept of the visualization. It may be a particular component of the interface, or something more abstracted such as a novel interaction method or an underlying data structure. We use this component of the sheet to visually describe, plan and detail the parti, the visual form should be appropriate for its nature. For example, we could use this section to create an expanded, or zoomed image of one component of the GUI, or use a flow diagram to describe how the user would interact with the data.
5. **Discussion.** Discuss advantages and disadvantages of this sketch. I.e., critique the design for its fit to task or suitability, potential and novelty, feasibility of implementation, scalability and extensibility, and clarity. This final stage of the sheet could be undertaken with the client.

These three designs then can aid client discussions (or can be used on own in self-reflection). When discussing the designs it is preferable to explain the process that is being followed. Developers should briefly describe the FdS methodology and explain the three ideas (sheets 2,3

and 4), describe the parti and explain what operations could be possible, and allow for discussion of the ideas.

3.5 FdS Sheet 5: Realization

The final sheet is the realization design (see Fig. 3). This is what the developer considers to be the final concept before delivery. It contains enough information to outline what it looks like, how users operate it, what its main idea is, and enough detail to implement it. The design on this sheet can be the same as one on sheet 2,3 or 4. Or it can be an amalgamation of the concepts. Indeed, the parts are the same (except the last). For completeness, they are: Information, Layout, Operations, Focus/Parti and finally *Detail*.

The extra detail that is included here could include:

- **Description** of the main algorithms, including any references. Detail of any design patterns used, or data structures proposed.
- **Maths** underpinning the idea and any calculations. Other measurements and plans of positioning of parts of the screen or estimations of proportion.
- **Dependencies** and requirements of the software, e.g., libraries that the software is built with.
- **Estimates** of cost, time or man-months of effort.
- Any **other** requirements of hardware or software, or pixel/screen size etc.

3.6 FdS resources

We encourage users to use a single (large) piece of paper for each sheet. The use of a single sheet per stage is important, because it brings together all the concepts in one place. This becomes a ‘unit’ with different parts and views on the same idea. I.e., the one sheet contains a sketch of the interface, a glimpse into the main parts and some important algorithms (the parti), user operations, and discussion over advantages and disadvantages. It is both convenient and sensible to keep all this information together. Where it may be useful to use more physical sheets is for ‘sheet 1’. We have experience of using several physical pages to explore different ideas, we believe it is better to think through many different possible solutions at the beginning of the process and refine them.

We also encourage A3 drawing paper. This gives a larger physical area to draw, where users are less concerned about space, and provides more freedom to be creative. Drawing paper is acid free, thicker, and longer lasting, etc. However students have also been successful with cheaper A3 printer paper. A3 pages can be scanned and printed to A4 and still be read. There are many different types of pens and pencils that can be used. Users obviously would have their own preferences; we encourage users to purchase two black fade-resistant drawing-pens (0.8 and 0.5 nibs), and a few color pens (e.g., red, green, blue and yellow). A ruler can also be useful. We encourage pens, and only a few colors, because we do not want users to get distracted by the medium and/or waste time choosing colors or rubbing out ideas, rather to focus on ideation.

4 EXPERIENCE WITH THE FdS

The FdS can be used in different situations. We demonstrate its use through three case-studies: (1) a usable tool for designers; (2) a design facilitation technique for industry, and (3) a teaching methodology for educators.

4.1 FdS Usability

To evaluate the FdS we used a modified System Usability Scale (SUS) questionnaire [5]. Our hypothesis was that the FdS is easy to use and helps designers to ideate and design information visualization tools. The SUS is suitable because it is a simple questionnaire, easy to administer, can be used on small sample sizes with reliable results, used by many researchers, adapted for different systems, and demonstrated to differentiate between usable and unusable systems [2].

Because the FdS is not an interactive system, we modified the questionnaire by changing the word ‘system’ with ‘method’, and ‘functions’ to ‘parts’, as follows:

1. I think that I would like to use this method frequently
2. I found the method unnecessarily complex
3. I thought the method was easy to use
4. I think that I would need the support of a technical person to be able to use this method
5. I found the various parts in this method were well integrated
6. I thought there was too much inconsistency in this method
7. I would imagine that most people would learn to use this method very quickly
8. I found the method very cumbersome to use
9. I felt very confident using the method
10. I needed to learn a lot of things before I could get going with this method

A five point Likert scale is used on the questions. After administering this questionnaire we measured the reliability to have a high Cronbach's alpha ($\alpha=0.8$), which indicates that alternation of the questionnaire did not affect the reliability of the scale. We surveyed 22 students, who had completed the Information Visualization module in the past 3 years; 53 students have completed the module since 2010 (42% response rate). One student miss-completed the questionnaire and we omitted their responses from the calculation.

The average SUS score was 74.29 (see Fig. 7), where the FdS was perceived to have 'good' usability. The term 'good' corresponds to the adjective scale presented by Bangor et al. [2]. While the scale by Bangor et al. concerns interactive systems, we believe it can be used as a comparison measure despite the 'pen-and-paper' nature of the FdS methodology.

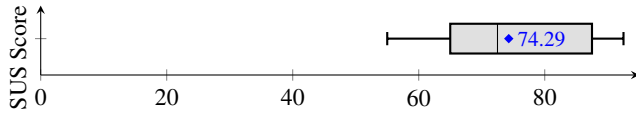


Fig. 7: Box plot of SUS scores with whiskers for minimum and maximum scores. The average score is 74.29, indicating 'good' usability in the adjective scale presented by Bangor et al. [2].

Table 1: Days of work for FdS vs. Conventional Approach.

Method	Designs	Days
FdS	5.5	14.1
CA	10.5	25.1

4.2 FdS in the Industry

The FdS method was used in a commercial setting, over a period of three years (2011-14), in parallel with a second, more conventional design approach, commonly used in the industry. The company designer learned of the FdS methodology in late 2011 [36], receiving a copy from a student volunteer who had attended the Eurographics conference. Recently, in collaboration with the authors, the company has evaluated the data and reflected on the process. This company represents a typical SME. They are a small firm who take on design and implementation work. The experience gained by this use-case can therefore be applied to other situations.

Our use-case focuses on the design of web application interfaces that have an interactive data visualization element. Our goal is not to propose the FdS as a web-design methodology, but to highlight its versatility for designing data visualization elements in a variety of applications.

Their *conventional approach* (CA) is an ongoing iterative process focusing on the design of GUI elements of a web application. CA involves a design phase, where wireframes of the GUI are produced based on the client's specification, and an evaluation phase, in which the client provides feedback. The GUI designs are refined in each iteration until the client is satisfied.

Over those three years the FdS method has been used with 9 clients, and the CA with 12. Table 1 compares the data from both the FdS and

CA. The 'Designs' column contains the average number of actual concepts generated before the client approved the design for each method. It is important to note that for the FdS column the number of designs refers to the number of sheets, not instances of the FdS methodology. On a number of occasions a 6th (and on one occasion a 7th) sheet was required (a revised realisation sheet) to meet new client requirements, or to accommodate revised assumptions resulting in an average of 5.5 sheets. The 'Days' column refers to the average length of time that each method took from initial concept to client approval.

Overall, clients approved the FdS designs quicker than they did with CA. Any alterations on designs produced with the FdS were typically approved during a meeting. However, with CA the client usually preferred to have additional time to consider proposed changes, extending the feedback phase over multiple days.

We believe that this extended feedback period was not due to the quality of the final concepts (as these were both implemented by the same designer), but due to the structure of the approach. The FdS begins by generating a wide range of concepts, though brainstorming, and conceptualizing. These are then filtered and refined to produce three good quality designs, before a final proposition is produced. At each stage of refinement there is the opportunity to explore various options along with the client, allowing them to mentally visualize, consider and discard various different configurations, without the need to implement them as a full concepts. The traditional method by comparison is linear, so, for the client to see multiple configurations they must explicitly request them during feedback. Consequently, the final designs typically end up being variations on a single theme.

4.3 FdS in Teaching – focusing on Masters module on Information Visualization

The FdS has been used to teach information visualization design at different levels and for several types of courses. These include an Information Visualization module as part of an Advanced Computer Science Masters qualification, with computer science major students in their final year computing project (in their design chapter), and a web-technologies class for students majoring in Information Systems.

In this section we principally focus on our experience with the FdS in teaching Information Visualization on the Advanced Computer science master program. Students take lectures, practical tutorial sessions and an end-of-semester written examination. The assessment is structured around the FdS, with students planning and ideating alternative designs, before implementing a visualization tool.

4.3.1 The task

For the coursework assessment the students, with advice from their tutor, first locate an open dataset (e.g., from Data.Gov, Guardian). By choosing their own dataset students can help each other without problems of plagiarism and follow a topic they find interesting.

The students are given an introduction on the FdS methodology, a live demonstration of its use and then they evaluate their chosen data. They perform a component analysis [4] where they analyze the components of the data and write down notes of this process, looking at variables, types, categories, temporal and range (see Sec. 3.2). They are also asked to consider a task for the user to perform; (when using the FdS with a client the task would evolve through discussion).

During a two hour practical session the students complete the FdS sheets. Subsequently they were given formative feedback on their designs and how they had followed the process. They could then change their designs (based on any formative comments), scan their sheets and submit the FdS as a PDF document to be marked.

4.3.2 The students and their learning

We have used this strategy in the Information Visualization module since 2010 and have recorded FdS designs implementations and their results for five years. We have gathered FdS sheets, tools built and marks given for 53 students (about 10 students per year). 58% of the students were from the UK and first language English with the rest mostly from Europe, India and middle East. No correlation to nationality was found ($r = 0.0206, n = 53$) with their results and performance

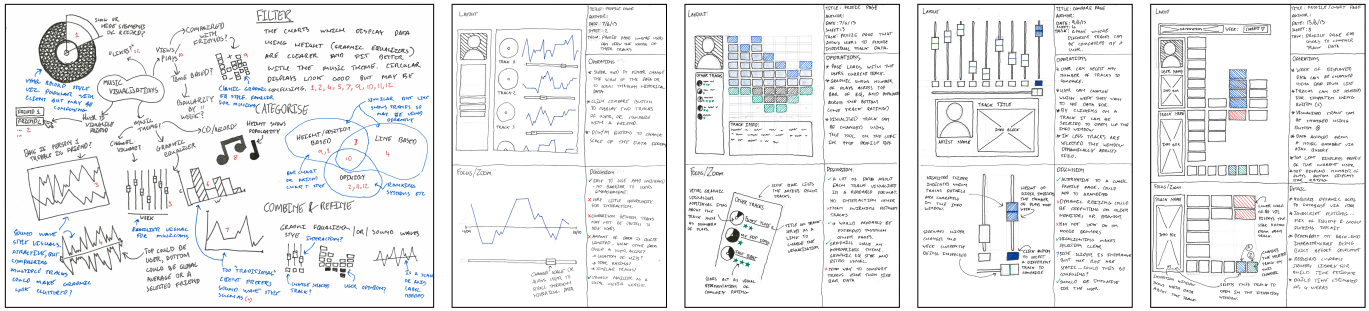


Fig. 8: An example of FdS produced in a commercial setting, depicting a concept for a music-based social media website, where amateur musicians can upload and share music they had produced. The website would then track, and visualize various data about their songs on their profile page, as well as maintaining a global ‘chart’ across all the users on the network.

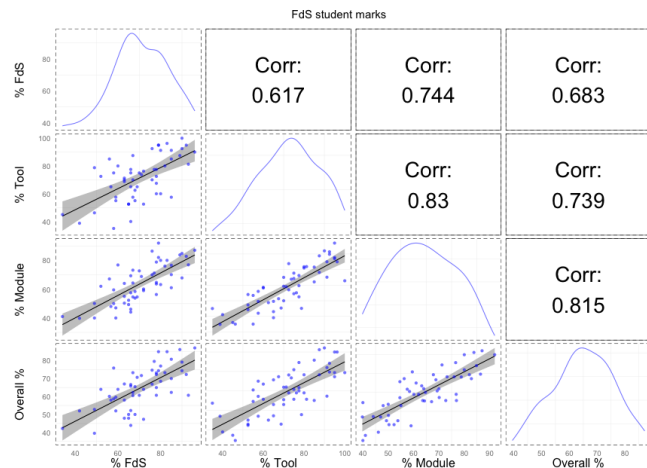


Fig. 9: We compare the mark given to the 53 students between 2010 and 2015 for the FdS, Tool that they built, whole Module and their Overall percentage of their masters qualification. Note the strong Pearson correlation with Module and overall percentage.

throughout the module. The students came to the module with no specific knowledge of information visualization, but all had programming experience. None of them had done the FdS before taking the class.

4.3.3 Evaluation as a teaching methodology

Fig. 9 demonstrates the results of the marks. Each sheet was marked out of 10, with 2 marks allocated for each FdS part. The overall score was created from an average of the 5 sheets. The Module percentage is taken from a weighting of their FdS, Tool and presentation, and examination result. At the end of the module the students demonstrated their tools and discussed what they had discovered in the data. They were marked on how they told their story, on the effectiveness of their mapping, whether they followed good practice, and which interactive capabilities were included in the visualizations. This marking scheme was announced to the students at the start of the module. To evaluate its effectiveness as an instrument for teaching, we discuss three questions:

Is it suitable, and does it correlate with other marks? When looking at students’ grades, it is often useful to correlate grades of a particular exercise with their averages across all other exercises. This gives the educator an understanding whether marks for a specific exercise are congruent to other aspects of their course. While there are many factors that effect student performance (such as attitude, engagement, belief of own design skills) usually students perform similarly in all aspects. Consequently the correlation gives the educator confidence that the assessment is suitable and students performed it to the best of their ability. Looking at the statistics (see Fig. 9) the FdS

correlates closely with their module average ($r = 0.815, n = 53$) and reasonably strong with their overall marks ($r = 0.683, n = 53$). These indicate that this is a suitable exercise and (as expected) good students perform well, and poorer students do (on average) worse. The lower correlation of the FdS with their overall score may, however, point to something else: that this is a unfamiliar task. We are asking students to think divergently, and divergent thinking is not a crosscutting program objective in computer science courses (we expand this idea below).

Does it help students plan? As part of our end of Semester module-evaluation process, we asked students to comment on the FdS in connection with the taught element of the module. This meant that their comments were anonymous and received as part of a wider process of evaluating the module as a whole. In this regard, students wrote positively: “The tools [programming and FdS design] that this course armed me with will help in any future task that required the visualization of information. It will also be useful in the writing of future papers and my masters thesis”. “FdS is a great methodology to use for quick ideation that is also very practical”. “...it requires very little introduction, and possibly the best tool to use in discussions with clients who are often not versed in the technological aspects of a project.”

Both undergraduate and taught postgraduate students do individual projects. They all consider and write-about alternative designs. However, students always struggled how to organize a chapter on ‘design’. Consequently, we have been using the FdS to help students contemplate alternative designs, and also structure this chapter. I.e., each subsection explains a page of the FdS. Following the FdS has helped students plan solutions, organize their thoughts and commit ideas to paper. One student wrote “It added a lot of structure to an otherwise unstructured section. Usually my designs lack detail and just end up as a big mess of thoughts. Using the FdS I could put my ideas in order”.

Does it help students to learn about, and to think divergently? Divergent thinking is not an easy skill to learn. In fact, one student said “you are asking us to think thoughts that are unusual to us”. Maybe this is a poor reflection on the education system; indeed, various researchers have criticized computing programs (and in fact other degree subjects) with the lack of design in teaching. However, ‘design thinking’ is starting to expand (e.g., d.school in Stanford, design thinking at Open University, and Enterprise by design at Bangor). Likewise design is important for information visualization [51] and information visualization is important for designers [16].

There are several challenges often posited about modern computer science education, including the learners’ lack of experience with user-interface design, and the inexperience with processing vague goals or ill-defined questions asked by the educators [37]. This limits their ability to distill a clear specification. Design thinking [43] is being encouraged as a solution to develop these skills, and helps students learn to handle uncertain requirements or wicked problems [6]. The FdS is another tool to help teach this type of skill, and students to learn to manage and work through such ill-defined questions [37].

The lowest score on the FdS for the whole five year period was for 34%, and the highest was 96%. This lowest score was low because the student did not complete all five sheets. Students were allowed to

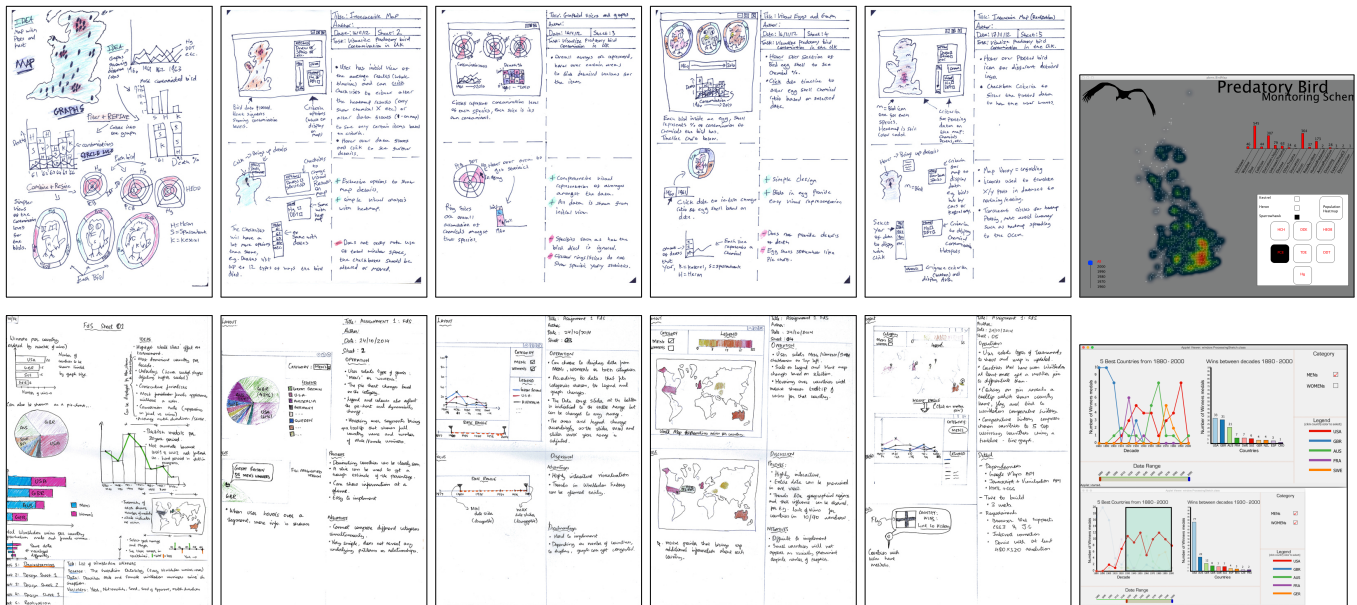


Fig. 10: Work from two students are shown. Both students were studying for their advanced Computer Science Masters (MSc). They performed the FdS as part of their Information Visualization coursework assessment. Each tool was implemented using Processing.org. These pictures show their five sheets (1 to 5, left to right) followed by a screenshot of their tool.

re-submit the FdS within two weeks of receiving their marks, but with a penalty of a 10% deduction on their second mark. This particular low-scoring student resubmitted and achieved a non-penalized score of 68%. This demonstrates that the student substantially improved, through this iteration.

To demonstrate how students develop ideas and work through different solutions, we provide two typical examples of the 53 students' work, see Fig. 10. (1) The first example shows data of predatory birds in the UK. The data contains information about breeding pairs, deaths of birds and population growth. The student worked through several ideas, including temporal, circular and map-based visualization; developing their main ideas in sheets 2,3,4 and eventually focusing on a heatmap visualization for their realization design. While there are aspects that could be improved with their implementation, the student has clearly considered different ideas, and has created a working and interactive visualization of this data. (2) The second example is of Olympic medal wins. Two screenshots are included, upper shows the full tool, with the lower view focusing on USA data. The student considered pie charts, stacked bar charts, map views through the process and finally focused on a timeline as the parti visualization. Through his tool he discovered the 'missing' Olympic games during WWII and provided interactivity to allow users to explore the data and compare medal wins per country.

5 DISCUSSION & CONCLUSIONS

There are many types of interfaces that software developers wish to create. We acknowledge that the FdS can be used for other types of interface design. In fact, we have used it to teach interactive visual tools in JavaScript on the web, and for a range of final-year undergraduate projects. However, this methodology works best when there is a visual interface to build, where a user can consider different visual depictions and layouts of the interface, as well as different interaction operations.

To make the FdS applicable for general interface design, obviously the preparation stage (Sec. 3.2) needs to be swapped for general requirements of the system and the parti needs to focus on the main idea of the interface (rather than the mapping of the data). But it should be possible to use the FdS methodology for many different interactive systems. From our experience, however, students struggle to use the FdS when the tool they are designing is simple. Systems that have a

single input and a single output file (or result) make the Operations panel moot. This is because in such systems, while the interface may be simple, the complexity is encapsulated within the underpinning algorithms and not a visual interface.

The FdS design methodology provides a structured method for users to think about ideas and sketch design solutions. The strategy has been successfully evaluated using the SUS methodology, demonstrating a 'good' level of usability (according to the scale by Bangor et al. [2]). It has been used by a developer at an SME and demonstrated to save time, and therefore money for that company. Shorter turn-around times were achieved. Finally we have explained its use in teaching, and have described how students have used the method in the planning of an information visualization tool of chosen data, and also in individual projects (and the presentation of alternative designs in their design chapter).

The FdS assumes that the user knows about their data. We refer the reader to Sec. 3.2 where we included guidance on how to evaluate the data, as well as to the component analysis model of Bertin [4]. In addition, we have not focused on effective mapping of data, nor on interaction design, because other authors focus their attention to this challenge, (e.g., [4, 53, 56]). However, we do believe that there is a need to have better guidelines on these tasks (including, data analysis, data mapping, interaction design) especially for learners.

The methodology encourages divergent thinking and enables developers and clients to discuss designs and iterate alternative solutions. The FdS is another tool in the developers' toolbox. The simplicity yet comprehensiveness of the FdS means that it is a useful asset in the ideation and design of interactive information visualization applications.

ACKNOWLEDGMENTS

We acknowledge the students of the Information Visualization module, we thank the VASS workshop participants and industrial partners. Without these individuals, and their enthusiasm, the FdS would not be such a success. We acknowledge conversations with Andy Goodman (Enterprise by Design, Bangor) at the inception of the FdS, and feedback from colleagues in Bangor (particularly Rick Walker, Steve Marriott and Nigel John). The European Commission partially funded this research through project EVIVA, 531140-LLP1-2012-1-UKKA3-KA3MP in the Lifelong Learning Programme.

REFERENCES

- [1] G. Andrienko, N. Andrienko, U. Demsar, D. Dransch, J. Dykes, S. I. Fabrikant, M. Jern, M.-J. Kraak, H. Schumann, and C. Tominski. Space, time and visual analytics. *Int. J. Geogr. Inf. Sci.*, 24(10):1577–1600, 2010.
- [2] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *Int J Hum-Comput Int*, 24(6):574–594, 2008.
- [3] M. Baskinger and K.-C. Nam. Visual narratives: the essential role of imagination in the visualization process. In *Proc. Asia-Pacific Symp. on Info. Vis. – Volume 60*, APVis '06, pages 217–220, Darlinghurst, Australia, 2006. Australian Computer Society, Inc.
- [4] J. Bertin. *Semiology of graphics*. University of Wisconsin Press, 1983.
- [5] J. Brooke. SUS – a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.
- [6] R. Buchanan. Wicked problems in design thinking. *Design issues*, 8(2):5–21, 1992.
- [7] B. Buxton. *Sketching user experiences: getting the design right and the right design*. Morgan Kaufmann, 2010.
- [8] M. Chen and H. Jäenicke. An information-theoretic framework for visualization. *IEEE Trans. Vis. Comput. Graphics*, 16(6):1206–1215, 2010.
- [9] E. H. Chi. A taxonomy of visualization techniques using the data state reference model. In *Proc. Info. Vis.*, pages 69–75, Washington, DC, USA, 2000. IEEE Computer Society.
- [10] D. Council. Eleven lessons: managing design in eleven global companies desk research report, 2007. www.designcouncil.org.uk.
- [11] B. Craft and P. Cairns. Sketching sketching: outlines of a collaborative design method. In *Proc. British HCI*, pages 65–72, Swindon, UK, 2009. British Computer Society.
- [12] F. Crick. The impact of linus pauling on molecular biology. In *The Pauling Symposium: Special Collections, The Valley Library, Oregon State University, OR*, 1996.
- [13] G. Curtis and L. Vertelney. Tutorial 33, storyboards and sketch prototypes for rapid interface visualization. In *CHI '90*, Seattle, Washington, 1990.
- [14] E. De Bono. *Lateral Thinking: Creativity Step by Step*. Panguin, 2009.
- [15] E. De Bono. *Six Thinking Hats*. Penguin UK, 2009.
- [16] G. Dove, S. Jones, J. Dykes, A. Brown, and A. Duffy. Using data visualization in creativity workshops: A new tool in the designer's kit. In *Proc. Creativity & Cognition, C&C '13*, pages 304–307. ACM, 2013.
- [17] M. Frederick. *101 Things I Learned in Architecture School*. MIT Press, September 2007.
- [18] B. Gaver and H. Martin. Alternatives: exploring information appliances through conceptual design proposals. In *Proc. of SIGCHI*, pages 209–216, New York, NY, USA, 2000. ACM.
- [19] M. Harrower and C. A. Brewer. Colorbrewer.org: An online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.
- [20] J. Heer and M. Agrawala. Software design patterns for information visualization. *IEEE Trans. Vis. Comput. Graphics*, 12(5):853–860, Sept 2006.
- [21] S. Heller and R. Landers. *Infographics Designers' Sketchbooks*. Princeton Architectural Press, 2014.
- [22] W. Hibbard, C. Dyer, and B. Paul. A lattice model for data display. In *IEEE Proc. Vis.*, pages 310–317, Oct 1994.
- [23] B. Jackson, D. Coffey, L. Thorson, D. Schroeder, A. M. Ellingson, D. J. Nuckley, and D. F. Keefe. Toward mixed method evaluations of scientific visualizations and design process as an evaluation tool. In *Proc. BELIV Workshop: Beyond Time and Errors-Novel Evaluation Methods for Visualization*, page 4, New York, NY, USA, 2012. ACM.
- [24] T. Jankun-Kelly, K.-L. Ma, and M. Gertz. A model and framework for visualization exploration. *IEEE Trans. Vis. Comput. Graphics*, 13(2):357–369, 2007.
- [25] S. Johnson. *Where good ideas come from: The natural history of innovation*. Penguin UK, 2010.
- [26] D. F. Keefe, D. Acevedo, J. Miles, F. Drury, S. Swartz, and D. Laidlaw. Scientific sketching for collaborative vr visualization design. *IEEE Trans. Vis. Comput. Graphics*, 14(4):835–847, July 2008.
- [27] G. Klein, B. Moon, and R. Hoffman. Making sense of sensemaking 2: A macrocognitive model. *IEEE Intelligent Sys.*, 21(5):88–92, Sept 2006.
- [28] J. Lansdown and S. Schofield. Expressive rendering: A review of nonphotorealistic techniques. *IEEE Comput. Graph. Appl.*, 15(3):29–37, 1995.
- [29] S. McKenna, D. Mazur, J. Agutter, and M. Meyer. Design activity framework for visualization design. *IEEE Trans. Visual. Comput. Graphics*, 20(12):2191–2200, Dec 2014.
- [30] T. Munzner. A nested process model for visualization design and validation. *IEEE Trans. Visual. Comput. Graphics*, 15:921–928, Nov 2009.
- [31] P. Pirolli and S. Card. The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis. In *Proc. Intelligence Analysis*, volume 5, pages 2–4, 2005.
- [32] H. C. Purchase, N. Andrienko, T. Jankun-Kelly, and M. Ward. Theoretical foundations of information visualization. In *Information Visualization*, pages 46–64. Springer, 2008.
- [33] M. Rettig. Prototyping for tiny fingers. *Commun. ACM*, 37(4):21–27, 1994.
- [34] D. Roam. *The back of the napkin: Solving Problems and Selling Ideas with Pictures*. Portfolio, 2008.
- [35] J. C. Roberts. State of the Art: Coordinated & Multiple Views in Exploratory Visualization. In G. Andrienko, J. C. Roberts, and C. Weaver, editors, *Proc. Int. Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV2007)*, pages 61–71. IEEE Computer Society Press, July 2007.
- [36] J. C. Roberts. The Five Design-Sheet (FdS) approach for Sketching Information Visualization Designs. In S. Maddock and J. Jorge, editors, *Proc. Eurographics 2011 – Education Papers*, pages 27–41. The Eurographics Association, 2011.
- [37] J. C. Roberts, D. Keim, T. Hanratty, R. Rowlingson, R. Walker, M. Hall, Z. Jacobson, V. Lavigne, C. Rooney, and M. Varga. From Ill-defined Problems to Informed Decisions. In M. Pohl and J. Roberts, editors, *EuroVis Wkshp Visual Analytics*, pages 7–11. Eurographics, 2014.
- [38] J. C. Roberts, J. Yang, O. Kohlbacher, M. Ward, and M. Zhou. Novel visual metaphors for multivariate networks. In A. Kerren, H. Purchase, and M. Ward, editors, *Multivariate Network Visualization*, volume 8380 of LNCS, pages 127–150. Springer International Publishing, 2014.
- [39] Y. Rogers, H. Sharp, and J. Preece. *Interaction Design: Beyond Human-Computer Interaction*. John Wiley and Sons Ltd, 2007.
- [40] E. Sanders and P. Stappers. Co-creation and the new landscapes of design. *CoDesign*, 4(1):5–18, 2008.
- [41] M. Sedlmair, M. D. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2431–2440, Dec 2012.
- [42] E. Shen, S. Li, X. Cai, L. Zeng, and W. Wang. Sketch-based interactive visualization: a survey. *Journal of Visualization*, 17(4):275–294, 2014.
- [43] B. Shneiderman. Creating creativity: User interfaces for supporting innovation. *ACM Trans. Comput.-Hum. Interact.*, 7(1):114–138, Mar. 2000.
- [44] R. Spence. *Information visualization: An introduction (Third Edition)*. ACM Press/Addison Wesley, 3rd edition, 2014.
- [45] M. Tory and T. Moller. Rethinking visualization: A high-level taxonomy. In *IEEE Symp. Info. Vis. 2004.*, pages 151–158. IEEE, 2004.
- [46] M. Tovey, S. Porter, and R. Newman. Sketching, concept development and automotive design. *Design Studies*, 24(2):135 – 153, 2003.
- [47] E. R. Tufte and P. Graves-Morris. *The visual display of quantitative information*, volume 2. Graphics press Cheshire, CT, 1983.
- [48] J. Uglow. *Lunar Men: The Friends Who Made the Future*. Faber and Faber, 2003.
- [49] C. Upson, T. Faulhaber, D. Kamins, D. Schlegel, D. Laidlaw, F. Vroom, R. Gurwitz, and A. vanDam. The application visualization system: A computational environment for scientific visualization. *IEEE Comput. Graph. Appl.*, 9(4):30–42, 1989.
- [50] J. van Wijk. Views on visualization. *IEEE Trans. Vis. Comput. Graph.*, 12(4):421–432, 2006.
- [51] A. Vande Moere and H. Purchase. On the role of design in information visualization. *Information Visualization*, 10(4):356–371, Oct. 2011.
- [52] J. Walny, S. Huron, and S. Carpendale. An exploratory study of data sketching for visual representation. *Computer Graphics Forum*, 34(3):231–240, 2015.
- [53] C. Ware. *Information visualization (Interactive Technologies)*. MK, 2012.
- [54] R. Wiseman. *59 seconds: think a little, change a lot*. Pan, 2010.
- [55] J. Wood, P. Isenberg, T. Isenberg, J. Dykes, N. Boukhelifa, and A. Slingsby. Sketchy rendering for information visualization. *IEEE Trans. Vis. Comput. Graph.*, 18(12):2749–2758, dec. 2012.
- [56] J. S. Yi, Y. ah Kang, J. T. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. Vis. Comput. Graph.*, 13(6):1224–1231, Nov 2007.
- [57] J. W. Young. *A Technique for Producing Ideas*. McGraw-Hill, 2003.
- [58] C. Ziemkiewicz and R. Kosara. The shaping of information by visual metaphors. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1269–1276, 2008.